Worksheet 3 Binary arithmetic Answers

Task 1

Carry out the following binary sums showing your working out:

a) $101_2 + 111_2$

	1	1	
	1	0	1
	1	1	1
1	**1**	**0**	**0**

b) $10110_2 + 10111_2$

1		1	1		
	1	0	1	1	0
	1	0	1	1	1
1	**0**	**1**	**1**	**0**	**1**

c) $11_2 + 100001_2 + 101_2$

		1	1	1	
				1	1
1	0	0	0	0	1
			1	0	1
1	**0**	**1**	**0**	**0**	**1**

d) $10101_2 + 111011_2 + 1001_2$

1	1	1	1	1	1	
		1	0	1	0	1
	1	1	1	0	1	1
			1	0	0	1
1	**0**	**1**	**1**	**0**	**0**	**1**

Show how the following values can be stored as binary bytes within a computer system and determine the answer that would be calculated and stored:

e) $12_{10} + 13_{10}$

128	**64**	**32**	**16**	**8**	**4**	**2**	**1**
			1	1			
0	0	0	0	1	1	0	0
0	0	0	0	1	1	0	1
0	**0**	**0**	**1**	**1**	**0**	**0**	**1**

f) $174_{10} + 255_{10}$

							1	1	1	1	1	1	1		
0	0	0	0	0	0	0	0	1	0	1	0	1	1	1	0
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**0**	**1**	**0**	**1**	**1**	**0**	**1**

g) $19_{10} + 66_{10} + 74_{10}$

1					1		
0	0	0	1	0	0	1	1
0	1	0	0	0	0	1	0
0	1	0	0	1	0	1	0
1	**0**	**0**	**1**	**1**	**1**	**1**	**1**

Answer the following question:

h) A computer has been designed to work only in single bytes of data. Describe the problem that will be encountered when carrying out the sum $01111001_2 + 11111001_2$ if the answer is only allocated one byte of storage.

- The answer is 101110010 which is 9 bits
- A byte is 8 bits so answer is too long to store and byte would be 01110010
- Calculation results in an overflow error indicating the resulting 8 bits have an incorrect answer and should be discarded

Task 2 Binary subtraction **Answers**

Convert these decimal values into two's complement binary bytes:

a) -50_{10} = 11001110

b) -120_{10} = 10001000

c) 127_{10} = 01111111

d) -128_{10} = 10000000

e) Show that -50_{10} gives the same result as in (a) above using the following alternative method:

Assume the left-most bit represents -128. Start at -128 and then add the remaining values:

-128	64	32	16	8	4	2	1
1	**1**	**0**	**0**	**1**	**1**	**1**	**0**

Carry out the following calculations in two's complement binary bytes:

f) $-50_{10} - 30_{10}$ *or* $-50_{10} + -30_{10}$

1			1	1	1		
1	1	0	0	1	1	1	0
1	1	1	0	0	0	1	0
1	**0**	**1**	**1**	**0**	**0**	**0**	**0**

g) $-66_{10} + 34_{10}$

	1	1	1	1	1		
1	0	1	1	1	1	1	0
0	0	1	0	0	0	1	0
1	**1**	**1**	**0**	**0**	**0**	**0**	**0**

h) $-88_{10} - 12_{10}$

1	1						
1	0	1	0	1	0	0	0
1	1	1	1	0	1	0	0
1	**0**	**0**	**1**	**1**	**1**	**0**	**0**

i) $22_{10} - -14_{10}$

		1	1	1	1		
0	0	0	1	0	1	1	0
0	0	0	0	1	1	1	0
0	**0**	**1**	**0**	**0**	**1**	**0**	**0**

Task 3 Fixed point binary **Answers**

Convert these decimal values into a fixed-point binary byte where the first 4 bits represent the whole number part and the the last 4 bits represent the fractional part:

a) 8.5_{10} = 10001000

b) 14.25_{10} = 1110 0100

c) 0.125_{10} = 00000010

d) 5.5625_{10} = 01011001

e) 1.9375_{10} = 00011111

f) 0.0625_{10} = 00000001

Convert these fixed-point binary bytes to decimal where the first 3 bits represent the whole number part and the the last 5 bits represent the fractional part:

g) 10101010_2 = 5.3125

h) 10111011_2 = 5.84375

i) 00111011_2 = 1.84375

j) 11111000_2 = 7.75

k) 00011111_2 = 0.96875

l) 11111111_2 = 7.96875